

S. Y. Pidopryhora

V. V. Choplyak

THE ROLE OF ENGLISH IN CODE READABILITY AND SOFTWARE DOCUMENTATION

Vinnytsia National Technical University

Анотація

Публікація досліджує роль англійської мови як фундаментального елемента забезпечення читабельності програмного коду в сучасній програмній інженерії. Аналізується вплив лексичної семантики англійської мови, граматичних структур та конвенцій іменування ідентифікаторів на розуміння, підтримку та модифікацію програмного забезпечення. Особливу увагу приділено концепції самодокументованого коду (self-documenting code) та міжнародним стандартам іменування. Результати дослідження демонструють, що використання семантично точних англомовних ідентифікаторів значно підвищує читабельність коду, зменшує когнітивне навантаження на розробників і сприяє ефективній командній взаємодії.

Ключові слова: читабельність коду, конвенції іменування, програмна інженерія, англійська мова в IT, самодокументований код, семантика.

Abstract

The publication examines the role of the English language as a fundamental element in ensuring code readability in modern software engineering. The influence of English lexical semantics, grammatical structures, and identifier naming conventions on software comprehension, maintenance, and modification is analyzed. Special attention is given to the concept of self-documenting code and internationally accepted naming standards. The study demonstrates that semantically precise English identifiers significantly enhance code readability, reduce cognitive load, and facilitate effective collaboration among developers.

Keywords: code readability, naming conventions, software engineering, English for IT, self-documenting code, semantics.

Introduction

In modern software engineering, source code serves not only as a sequence of formal instructions executed by a computer but also as a primary means of communication between developers. During the software lifecycle, code is read far more frequently than it is written, particularly during debugging, maintenance, refactoring, and collaborative development. Therefore, code readability is a critical quality attribute that directly affects software maintainability, reliability, and development efficiency [2].

Most contemporary programming languages, including Java, C++, Python, and JavaScript, are fundamentally based on English. Their syntax incorporates English keywords such as *class*, *return*, *if*, and *while*, while developers use English vocabulary to create identifiers representing data, operations, and system components. As a result, English functions not merely as a communication tool but as an integral structural and semantic component of source code.

According to established software engineering principles, particularly those described in the Clean Code methodology [1], meaningful naming is essential for creating readable and maintainable software. The clarity of identifiers directly influences how efficiently developers can understand and modify program logic.

Research results

Research in software engineering demonstrates that identifier naming conventions significantly influence program comprehension [3]. Meaningful English identifiers allow developers to understand the purpose and behavior of program elements without requiring additional explanation.

Consider the following example:

```
int x = 10;
int y = 5;
int z = x * y;
```

Although syntactically correct, this code lacks semantic clarity. A developer must analyze the surrounding context to understand its purpose.

In contrast, the following example demonstrates improved readability:

```
int itemPrice = 10;
int itemQuantity = 5;
int totalCost = itemPrice * itemQuantity;
```

Here, the identifiers clearly describe the meaning of each variable and the logic of the computation. This significantly improves comprehension and reduces the time required to interpret the code.

Method naming also plays a critical role in expressing program behavior:

```
void processData()
```

This name provides limited semantic information.

A more precise alternative improves clarity:

```
void calculateTotalCost()
```

This identifier clearly communicates the purpose of the method.

Such practices align with internationally accepted naming conventions, including camelCase for variables and methods and PascalCase for class names, which promote consistency and readability across software systems [4].

Analysis of the results

The analysis confirms that English serves as a semantic framework that enables developers to create self-documenting code. Self-documenting code is structured in such a way that its purpose and functionality can be understood directly from the identifiers and structure, reducing reliance on external comments or documentation.

From a cognitive perspective, meaningful English identifiers reduce the mental effort required to interpret program logic [3]. When identifiers clearly describe their purpose, developers can understand the system more quickly and accurately [2]. Conversely, poorly named identifiers increase cognitive load and make software more difficult to maintain.

This is particularly important in collaborative development environments, where developers frequently work with code written by others. In international teams, English serves as a universal semantic standard that ensures consistent understanding of software components. Clear and meaningful naming improves communication, reduces misunderstandings, and minimizes errors during modification.

Furthermore, precise English naming improves long-term maintainability. Software systems often evolve over time, and developers must regularly modify existing code. Semantically meaningful identifiers allow developers to understand system functionality even years after the original code was written [1].

These findings confirm that English proficiency directly contributes to software quality by improving readability, maintainability, and collaborative efficiency.

Conclusion

The English language plays a fundamental role in software engineering, functioning as both a communication medium and a structural component of source code. Meaningful English identifiers significantly enhance code readability, reduce cognitive complexity, and improve program comprehension. The use of semantically precise naming conventions enables developers to create self-documenting code, reducing dependence on external documentation and improving software maintainability.

Additionally, adherence to English-based naming standards facilitates effective collaboration in international development environments. Therefore, English proficiency should be considered a critical professional competency for software developers, as it directly influences software quality, maintainability, and overall effectiveness of software engineering processes.

REFERENCES

1. Martin, R. C. Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall, 2008.
2. dos Santos, F. F., et al. (2021). Evaluating Code Readability and Legibility: An Examination of Comprehension Tasks and Response Variables. *arXiv preprint arXiv:2110.00785*. Available at: <https://arxiv.org/abs/2110.00785>
3. Sankey, E. B. (2021). Code readability: A proposal on the effects of psychology and comprehension. *UNI ScholarWorks*. Available at: <https://scholarworks.uni.edu/hpt/1402>
4. Oracle Corporation. Java Code Conventions and Naming Standards.

Підопригора Світлана Юрївна – студентка групи ЗПІ-256, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: sveta26pipodo@gmail.com

Науковий керівник: **Чопляк Вікторія Володимирівна** – викладач англійської мови, кафедра іноземних мов, Вінницький національний технічний університет, м. Вінниця, e-mail: nikavnuchkova@gmail.com

Svitlana Y. Pidopryhora – Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: sveta26pipodo@gmail.com

Scientific Supervisor: **Victoria V. Choplyak** – English Language Instructor, Foreign Languages Department, Vinnytsia National Technical University, Vinnytsia, e-mail: nikavnuchkova@gmail.com